

## **LAPORAN TUGAS KECIL 1**

### **Penyelesaian Permainan Kartu 24 dengan Algoritma Brute Force**

Ditujukan untuk memenuhi salah satu tugas kecil Mata Kuliah IF2211 Strategi Algoritma (Stima) pada Semester II Tahun Akademik 2022/2023

Disusun oleh :

**Fajar Maulana H**

**13521080**



**PROGRAM STUDI TEKNIK INFORMATIKA  
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA  
INSTITUT TEKNOLOGI BANDUNG BANDUNG  
2023**

## Daftar Isi

<i>A. Algoritma brute force</i> .....	3
<i>B. Source Program</i> .....	4
<i>C. Screenshots input dan output</i> .....	19
<i>D. Link Repository</i> .....	28

## A. Algoritma *brute force*

Algoritma *brute force* merupakan algoritma untuk memecahkan sebuah persoalan secara sederhana, langsung, dan jelas caranya. Algoritma *brute force* memiliki pendekatan yang *straight forward* atau lempang untuk memecahkan suatu persoalan. Pada umumnya algoritma *brute force* didasarkan pada pernyataan pada persoalan dan definisi atau konsep yang terlibat. Algoritma ini biasanya mencoba semua kemungkinan jawaban yang ada pada sebuah persoalan. Kelebihan dari algoritma ini yaitu cukup mudah untuk diimplementasi pada semua persoalan komputasi dan mudah untuk dipahami, sedangkan kelemahannya, yakni karena algoritma ini mencoba semua kemungkinan, algoritma ini bisa saja tidak cukup efisien untuk digunakan pada sebuah persoalan. Oleh karena itu, algoritma ini biasanya membutuhkan waktu lebih lama daripada yang lain.

Penyelesaian permainan kartu 24 sebagaimana diulas pada tugas kecil ini menggunakan algoritma *brute force*. Pada awalnya program akan membaca inputan dari user yaitu 4 angka/huruf yang terdiri dari A, 2, 3, 4, 5, 6, 7, 8, 9, J, Q, K. Selain itu, program dapat men-*generate* 4 angka/huruf sendiri secara random. Oleh karena itu, pengguna dapat memilih untuk memasukan *input* atau men-*generate* sendiri. Program akan melakukan validasi terhadap input dari user. Kemudian, dari input yang dimasukkan oleh user, program akan melakukan validasi terhadap inputan tersebut yaitu 4 huruf/angka yang terdiri dari A, 2, 3, 4, 5, 6, 7, 8, 9, J, Q, K. Setelah input valid, berikut adalah Langkah *brute force* yang digunakan dalam program saya.

Pertama-tama, program akan melakukan perhitungan permutasi pada 4 huruf/angka yang sudah dimasukkan sebelumnya sehingga jumlah dari permutasinya yakni  $4!$ . Kemudian, program akan menyimpan hasil permutasi tersebut ke dalam *vector of char*. Setelah itu, program juga akan melakukan perhitungan pada kemungkinan operasinya yang ada yaitu sejumlah 64 kombinasi dan disimpan di dalam *vector of char*. Diharapkan dengan dilakukan pemisahan pada perhitungan komputasi, maka akan dapat mempercepat proses komputasi komputer itu sendiri sehingga lebih efisien. Lalu, dilakukan iterasi setiap 4 karakter pada *vector of char* hasil permutasi huruf/angka. Pada setiap 4 karakter tersebut akan dilakukan iterasi bagi setiap 3 karakter pada *vector of char* hasil kombinasi kemungkinan operasi, dengan demikian, pada akhirnya akan dihasilkan kemungkinan sebesar 1536 ( $64 \times 4!$ ).

Kedua, Setelah ditemukan 1536 kemungkinan, setiap kemungkinannya akan berbentuk sebagai berikut

$$A \text{ operasi}_1 B \text{ operasi}_2 C \text{ operasi}_3 D$$

$A, B, C, D$  = salah satu kemungkinan permutasi 4 angka/huruf  
 $\text{operasi}_1, \text{operasi}_2, \text{operasi}_3$  = salah satu kombinasi kemungkinan

Salah satu contoh, saat kita menerima input 2, 3, 7, 8, kita akan mendapatkan kemungkinannya, yaitu

$$\begin{aligned}
 &2 + 3 + 7 + 8 \\
 &2 + 3 + 7 - 8 \\
 &2 + 3 - 7 - 8 \\
 &\dots
 \end{aligned}$$

Setelah itu, setiap kemungkinan akan dilakukan operasi matematika dengan kemungkinan urutan operasinya ada 6 ,yakni operasi 1 dilakukan pertama, operasi 2 dilakukan kedua, operasi 3 dilakukan ketiga dan seterusnya. Terdapat 2 kemungkinan operasi yang sama ,yakni operasi 1 dilakukan pertama, operasi 3 dilakukan kedua, dan operasi 2 dilakukan ketiga dengan operasi 3 dilakukan pertama, operasi 1 dilakukan kedua, dan operasi 2 dilakukan ketiga karena bentuk ( A op1 B ) op2 ( C op3 D ). Oleh karena itu, jumlah kemungkinannya menjadi 5. Dengan demikian, total kemungkinan yang dicoba berjumlah 7680 dari  $24 * 64 * 5$ .

Ketiga, Dari urutan operasi tersebut dapat dilakukan perhitungan matematika dengan menggunakan fungsi yang ada untuk membaca operasi matematika dan mengeluarkan hasil perhitungannya. Dari hasil operasi matematika tersebut, dicari yang menghasilkan 24. Operasi yang memenuhi kita masukkan ke dalam *vector of string*. Dari *vector of string* tersebut akan di-filter sehingga operasi yang sama tidak terulang kembali.

## B. Source Program

*Source code* program ditulis dalam bahasa pemrograman C++ dan terdiri dari 1 file header function.hpp dan 2 file C++ yaitu main.cpp dan function.cpp.

### function.hpp

```

#ifndef FUNCTION_HPP
#define FUNCTION_HPP
#include <string>
#include <iostream>
using namespace std;

int chartoint(string cc);
/* Mengubah string angka ke int */

void saveAnswer(vector <string> s, string namaFile, int n);
/* I.S vector s, string namaFile, dan int n terdefinisi */
/* F.S n dan elemen vector s tersimpan di namaFile.txt */

bool checkValid(string input);
/* Mengembalikan true jika input valid */

void inputUser(int (&result)[4]);
/* I.S result terdefinisi */
/* F.S result terisi dengan input user */

```

```

void randomCard(int (&result)[4]);
/* I.S result terdefinisi */
/* F.S result terisi dengan random card */

void inttocard(int in);
/* I.S in terdefinisi */
/* F.S in ditulis dalam bentuk kartu */

void permutasiKartu(int a, int kedalaman, vector<int> &result, int input[4]);
/* I.S a, kedalaman, result, dan input terdefinisi */
/* F.S result terisi dengan permutasi kartu */

char inttochar(int in);
/* Mengubah int ke char */

void kombinasiOperator(vector<char> &result);
/* I.S result terdefinisi */
/* F.S result terisi dengan kombinasi operator */

void hitung1(char op1, char op2, char op3, int a, int b, int c, int d, int &n, vector<string> &s);
/* I.S op1, op2, op3, a, b, c, d, n, dan s terdefinisi */
/* F.S n dan s terisi dengan hasil perhitungan dari ((( a op1 b ) op2 c ) op3 d ) */

void hitung2(char op1, char op2, char op3, int a, int b, int c, int d, int &n, vector<string> &s);
/* I.S op1, op2, op3, a, b, c, d, n, dan s terdefinisi */
/* F.S n dan s terisi dengan hasil perhitungan dari ( a op1 b ) op2 ( c op3 d ) */

void hitung3(char op1, char op2, char op3, int a, int b, int c, int d, int &n, vector<string> &s);
/* I.S op1, op2, op3, a, b, c, d, n, dan s terdefinisi */
/* F.S n dan s terisi dengan hasil perhitungan dari ( a op1 ( b op2 c ) ) op3 d */

void hitung4(char op1, char op2, char op3, int a, int b, int c, int d, int &n, vector<string> &s);
/* I.S op1, op2, op3, a, b, c, d, n, dan s terdefinisi */
/* F.S n dan s terisi dengan hasil perhitungan dari a op1 ( ( b op2 c ) op3 d ) */

void hitung5(char op1, char op2, char op3, int a, int b, int c, int d, int &n, vector<string> &s);
/* I.S op1, op2, op3, a, b, c, d, n, dan s terdefinisi */
/* F.S n dan s terisi dengan hasil perhitungan dari a op1 ( b op2 ( c op3 d ) ) */

int stringtoint(string cc);
/* Mengubah string angka ke int */

float hitung(char op, float a, float b);
/* Mengembalikan hasil perhitungan a op b */

void hapusSama(vector<string> &s, int &n);
/* I.S s dan n terdefinisi */

```

```
/* F.S s dan n terisi dengan elemen yang tidak sama */
```

```
bool isNumber(string s, vector <string> v);
```

```
/* Mengembalikan true jika s ada di v */
```

```
void readFile(int (&angka)[4]);
```

```
/* I.S angka terdefinisi */
```

```
/* F.S angka terisi dengan angka dari file */
```

```
#endif
```

## function.cpp

```
#include "function.hpp"
```

```
#include <sstream>
```

```
#include <cstdlib>
```

```
#include <vector>
```

```
#include <iostream>
```

```
#include <fstream>
```

```
using std::ofstream;
```

```
int chartoint(string cc){
```

```
    int n = 0;
```

```
    if (cc == "A"){
```

```
        n = 1;
```

```
    }
```

```
    else if (cc == "2"){
```

```
        n = 2;
```

```
    }
```

```
    else if (cc == "3"){
```

```
        n = 3;
```

```
    }
```

```
    else if (cc == "4"){
```

```
        n = 4;
```

```
    }
```

```
    else if (cc == "5"){
```

```
        n = 5;
```

```
    }
```

```
    else if (cc == "6"){
```

```
        n = 6;
```

```
    }
```

```
    else if (cc == "7"){
```

```
        n = 7;
```

```
    }
```

```
    else if (cc == "8"){
```

```
        n = 8;
```

```
    }
```

```

    else if (cc == "9"){
        n = 9;
    }
    else if (cc == "10"){
        n = 10;
    }
    else if (cc == "J"){
        n = 11;
    }
    else if (cc == "Q"){
        n = 12;
    }
    else if (cc == "K"){
        n = 13;
    }
    else{
        n = -1;
    }
    return n;
}

```

```

void inttocard(int in){
    if (in == 1){
        cout << "A";
    }
    else if (in == 2){
        cout << "2";
    }
    else if (in == 3){
        cout << "3";
    }
    else if (in == 4){
        cout << "4";
    }
    else if (in == 5){
        cout << "5";
    }
    else if (in == 6){
        cout << "6";
    }
    else if (in == 7){
        cout << "7";
    }
    else if (in == 8){
        cout << "8";
    }
    else if (in == 9){

```

```

        cout << "9";
    }
    else if (in == 10){
        cout << "10";
    }
    else if (in == 11){
        cout << "J";
    }
    else if (in == 12){
        cout << "Q";
    }
    else if (in == 13){
        cout << "K";
    }
}

bool checkValid(string input)
{
    //check if input is empty
    if (input.empty())
    {
        cout << "Input is empty" << endl;
        return false;
    }
    //check if each card is valid
    stringstream ss(input);
    string temp;
    int n = 0;
    while (ss >> temp)
    {
        if (chartoint(temp) < 0)
        {
            cout << "Masukkan Huruf dan Angka Tidak Valid" << endl;
            return false;
        }
        n++;
    }
    //check if input is only have 4 cards
    if (n != 4)
    {
        cout << "Input is not 4 cards" << endl;
        return false;
    }
    return true;
}

void inputUser(int (&result)[4])

```



```

{
    //input user
    string input;
    //check input if valid when input is not empty, only have 4 cards, and each card is valid
    do{
        cout << "Masukkan kartu yang ingin di cek: ";
        getline(cin, input);
    }
    while (!checkValid(input));
    cout << "Input valid" << endl;
    stringstream ss(input);
    string temp;
    int n = 0;
    while (ss >> temp)
    {
        result[n] = chartoint(temp);
        n++;
    }
}

int n, catat[10];
bool status[10];
void randomCard(int (&result)[4])
{
    //random card
    for (int i = 0; i < 4; i++)
    {
        result[i] = rand() % 13 + 1;
        if (i != 4){
            inttocard(result[i]);
            cout << " ";
        }
    }
    cout << endl;
}

void tulis(int kedalaman, vector<int> &result, int input[4]){

    if(kedalaman >= n){

        for (int i = 0; i < n; i++)
        {
            result.push_back(input[catat[i]-1]);
        }
    }else{
        for(int i = 1;i<=n;i++){
            if(status[i] == false){
                status[i] = true;
            }
        }
    }
}

```

```

        catat[kedalaman] = i;
        tulis(kedalaman+1, result, input);
        status[i] = false;
    }
}
}
}

void permutasiKartu(int a, int kedalaman, vector <int> &result, int input[4]){
    n = a;
    tulis(kedalaman, result, input);
}

char inttochar(int in){

    if (in == 1){
        return '+';
    }
    else if (in == 2){
        return '-';
    }
    else if (in == 3){
        return '*';
    }
    else if (in == 4){
        return '/';
    }
    else{
        return ' ';
    }
}

void kombinasiOperator(vector <char> &result){
    for (int i = 1; i <= 4; i++)
    {
        for (int j = 1; j <= 4; j++)
        {
            for (int k = 1; k <= 4; k++)
            {
                result.push_back(inttochar(i));
                result.push_back(inttochar(j));
                result.push_back(inttochar(k));
            }
        }
    }
}

float hitung(float a, float b, char c){

```

```

    if (c == '+'){
        return a+b;
    }
    else if (c == '-'){
        return a-b;
    }
    else if (c == '*'){
        return a*b;
    }
    else if (c == '/'){
        return a/b;
    }
    else{
        return 0;
    }
}

void hitung1(char op1,char op2,char op3,int a, int b, int c, int d, int &n,vector <string> &s){
    float hasil = hitung(hitung(hitung(a,b,op1),c,op2),d,op3);
    string s1 = "((((" + to_string(a) + op1 + to_string(b) + ")") + op2 + to_string(c) + ")") + op3 + to_string(d) + ")=" +
    to_string(hasil);
    if (hasil == 24){
        s1 = "(" + to_string(a) + op1 + to_string(b) + ")") + op2 + to_string(c) + ")") + op3 + to_string(d) ;
        n++;
        s.push_back(s1);
    }
}

void hitung2(char op1,char op2,char op3,int a, int b, int c, int d, int &n, vector <string> &s){
    float hasil = hitung(hitung(a,b,op1),hitung(c,d,op3),op2);
    string s1 = "(" + to_string(a) + op1 + to_string(b) + ")") + op2 + "(" + to_string(c) + op3 + to_string(d) + ")=" +
    to_string(hasil);

    if (hasil == 24){
        s1 = "(" + to_string(a) + op1 + to_string(b) + ")") + op2 + "(" + to_string(c) + op3 + to_string(d) + ")";
        n++;
        s.push_back(s1);
    }
}

void hitung3(char op1,char op2,char op3,int a, int b, int c, int d, int &n,vector <string> &s){
    float hasil = hitung(hitung(a,hitung(b,c,op2),op1),d,op3);
    string s1 = "(" + to_string(a) + op1 + "(" + to_string(b) + op2 + to_string(c) + ")") + op3 + to_string(d) + "=" +
    to_string(hasil);

```

```

    if (hasil == 24){

        s1 = "(" + to_string(a) + op1 + "(" + to_string(b) + op2 + to_string(c) + ")" + op3 + to_string(d) ;
        n++;
        s.push_back(s1);
    }
}

void hitung4(char op1,char op2,char op3,int a, int b, int c, int d, int &n,vector <string> &s){
    float hasil = hitung(a,hitung(hitung(b,c,op2),d,op3),op1);
    string s1 = "(" + to_string(a) + op1 + "(" + to_string(b) + op2 + to_string(c) + ")" + op3 + to_string(d) + ")=" +
to_string(hasil);
    if (hasil == 24){
        s1 = to_string(a) + op1 + "(" + to_string(b) + op2 + to_string(c) + ")" + op3 + to_string(d) + ")";
        n++;
        s.push_back(s1);
    }
}

void hitung5(char op1,char op2,char op3,int a, int b, int c, int d, int &n,vector <string> &s){
    float hasil = hitung(a,hitung(b,hitung(c,d,op3),op2),op1);
    string s1 = "(" + to_string(a) + op1 + "(" + to_string(b) + op2 + "(" + to_string(c) + op3 + to_string(d) + ")") +
to_string(hasil);

    if (hasil == 24){
        s1 = to_string(a) + op1 + "(" + to_string(b) + op2 + "(" + to_string(c) + op3 + to_string(d) + ")";
        n++;
        s.push_back(s1);
    }
}

int stringtoint(string cc){
    if (cc == "1"){
        return 1;
    }else if(cc == "2"){
        return 2;
    }else if(cc == "3"){
        return 3;
    }else if(cc == "4"){
        return 4;
    }else if(cc == "5"){
        return 5;
    }else if(cc == "6"){
        return 6;
    }else if(cc == "7"){

```

```

        return 7;
    }else if(cc == "8"){
        return 8;
    }else if(cc == "9"){
        return 9;
    }else if(cc == "10"){
        return 10;
    }else if(cc == "11"){
        return 11;
    }else if(cc == "12"){
        return 12;
    }else if(cc == "13"){
        return 13;
    }else{
        return 0;
    }
}

bool isNumber(string s, vector <string> v){
    for (int i = 0; i < v.size(); i++){
        if (s == v[i]){
            return true;
        }
    }
    return false;
}

void hapusSama(vector <string> &s, int &n){
    vector <string> temp ;
    for (int i = 0; i < s.size(); i++){
        if (!isNumber(s[i],temp)){
            temp.push_back(s[i]);
        }
    }
    n = temp.size();
    s = temp;
}

void readFile(int (&angka)[4]){
    string namaFile;
    ifstream myfile;
    string line;
    do{
        cout << "Masukkan nama file: ";
        cin >> namaFile;
        myfile.open ("../test/"+namaFile);
        getline(myfile,line);
    }while(myfile.is_open());
}

```

```

    if (checkValid(line)){
        cout << "Input valid" << endl;
        stringstream ss(line);
        string temp;
        int n = 0;
        while (ss >> temp)
        {
            angka[n] = chartoint(temp);
            n++;
        }
    }else{
        cout << "Input tidak valid" << endl;
    }
    myfile.close();
} while(!checkValid(line));
}

void saveAnswer(vector <string> s,string namaFile = "answer.txt", int n = 0){
    ofstream myfile;
    myfile.open ("../test/"+namaFile);
    myfile << "Jumlah kemungkinan = " << n << endl;
    for (int i = 0; i < s.size(); i++){
        myfile << s[i] << endl;
    }
    myfile.close();
}

```

## main.cpp

```

#include <iostream>
#include <string>
#include "function.hpp"
#include <sstream>
#include <vector>
#include <ctime>

using namespace std;

void mainGame(){
    int inputArr[4];
    vector <int> result;
    vector <char> resultOp;
    string caseNum;
    time_t start, finish;
    bool valid = false;
    do
    {
        cout << "Masukkan Input atau Random [1/2]: ";
    }

```

```

cin.clear();
cin >> caseNum;
if (caseNum == "1" || caseNum == "2")
{
    valid = true;
}else{
    cout << "Input tidak valid" << endl;
}
} while (!valid);
cin.ignore( std::numeric_limits<std::streamsize>::max(), '\n' );
if (caseNum == "1")
{
    bool status = false;
    do{
        cout << "File atau Keyboard [1/2]: ";
        cin >> caseNum;
        if(caseNum == "1" || caseNum == "2"){
            status = true;
        }else{
            cout << "Input tidak valid" << endl;
        }
        cin.ignore( std::numeric_limits<std::streamsize>::max(), '\n' );
    }while(!status);
    if(caseNum == "1"){
        readFile(inputArr);
    }else{
        inputUser(inputArr);
    }
}
else
{
    randomCard(inputArr);
}
for (int i = 0; i < 4; i++)
{
    cout << inputArr[i] << " ";
}
time(&start);
permutasiKartu(4,0,result,inputArr);
kombinasiOperator(resultOp);
int n = 0;
vector <string> s;
for (int i = 0; i < result.size(); i = i+4)
{
    for (int j = 0;j<resultOp.size();j = j+3){
        string input;

```

```

        input = to_string(result[i]) + " "+resultOp[j]+" "+ to_string(result[i+1])+" "+ resultOp[j+1]+"
"+to_string(result[i+2])+" "+ resultOp[j+2]+" "+ to_string(result[i+3]);
        // cout << input << endl;

hitung1(resultOp[j],resultOp[j+1],resultOp[j+2],stringtoint(to_string(result[i])),stringtoint(to_string(result[i+1])),stringtoint(to
_string(result[i+2])),stringtoint(to_string(result[i+3])),n,s);

hitung2(resultOp[j],resultOp[j+1],resultOp[j+2],stringtoint(to_string(result[i])),stringtoint(to_string(result[i+1])),stringtoint(to
_string(result[i+2])),stringtoint(to_string(result[i+3])),n,s);

hitung3(resultOp[j],resultOp[j+1],resultOp[j+2],stringtoint(to_string(result[i])),stringtoint(to_string(result[i+1])),stringtoint(to
_string(result[i+2])),stringtoint(to_string(result[i+3])),n,s);

hitung4(resultOp[j],resultOp[j+1],resultOp[j+2],stringtoint(to_string(result[i])),stringtoint(to_string(result[i+1])),stringtoint(to
_string(result[i+2])),stringtoint(to_string(result[i+3])),n,s);

hitung5(resultOp[j],resultOp[j+1],resultOp[j+2],stringtoint(to_string(result[i])),stringtoint(to_string(result[i+1])),stringtoint(to
_string(result[i+2])),stringtoint(to_string(result[i+3])),n,s);
    }
}
hapusSama(s,n);
time(&finish);
cout << endl << "=====" << endl;
cout << n << " solusi ditemukan" << endl;
cout << "=====" << endl;
cout << "Solusi:" << endl;
for (int i = 0; i < s.size(); i++)
{
    cout << s[i] << endl;
}
cout << "=====" << endl;
cout << "Waktu yang dibutuhkan: " << difftime(finish,start) << " detik" << endl;
bool status = false;
do{
    cout << "Simpan ke file [Y/N]: ";
    cin >> caseNum;
    if(caseNum == "Y" || caseNum == "N"){
        status = true;
    }else{
        cout << "Input tidak valid" << endl;
    }
    cin.ignore( std::numeric_limits<std::streamsize>::max(), '\n' );
}while(!status);
if(caseNum == "Y"){
    cout << "Masukkan nama file: ";
    string input;
    cin >> input;

```



[illegible]

```
        status = true;
    }else{
        cout << "Input tidak valid" << endl;
    }
}while(!status);
if (input == "Y")
{
    masihMain = true;
}
if(input == "N")
{
    masihMain = false;
}
}
```

### C. Screenshots input dan output

#### 1. Input Pengguna Keyboard (Angka Semua)

```
2 3 4 5

Masukkan Input atau Random [1/2]: 1
File atau Keyboard [1/2]: 2
Masukkan kartu yang ingin di cek: 2 3 4 5
Input valid
2 3 4 5
=====
40 solusi ditemukan
=====
Solusi:
2*((3+4)+5)
2*(3+(4+5))
2*((3+5)+4)
2*(3+(5+4))
2*((4+3)+5)
2*(4+(3+5))
2*((4+5)+3)
2*(4+(5+3))
2*((5+3)+4)
2*(5+(3+4))
2*((5+4)+3)
2*(5+(4+3))
((3-2)+5)*4
(3-(2-5))*4
((3+4)+5)*2
(3+(4+5))*2
((3+5)-2)*4
(3+(5-2))*4
((3+5)+4)*2
(3+(5+4))*2
4*((3-2)+5)
4*(3-(2-5))
((4+3)+5)*2
(4+(3+5))*2
4*((3+5)-2)
4*(3+(5-2))
4*((5-2)+3)
4*(5-(2-3))
((4+5)+3)*2
(4+(5+3))*2
4*((5+3)-2)
4*(5+(3-2))
((5-2)+3)*4
(5-(2-3))*4
((5+3)-2)*4
(5+(3-2))*4
((5+3)+4)*2
(5+(3+4))*2
((5+4)+3)*2
(5+(4+3))*2
=====
Waktu yang dibutuhkan: 0.0210959 detik
Simpan ke file [Y/N]: Y
Masukkan nama file: hasil.txt
Main Lagi? [Y/N]: N
fajarherawan@Fajars-MacBook-Pro bin %
```

## 2. Input pengguna keyboard (Huruf Semua)

```
A J Q K
S E L A M A T D A
A W A Y
Masukkan Input atau Random [1/2]: 1
File atau Keyboard [1/2]: 2
Masukkan kartu yang ingin di cek: A J Q K
Input valid
1 11 12 13
=====
32 solusi ditemukan
=====
Solusi:
(1*12)*(13-11)
1*(12*(13-11))
((1*13)-11)*12
(1*(13-11))*12
1*((13-11)*12)
(12*1)*(13-11)
12*((1*13)-11)
12*(1*(13-11))
(12/1)*(13-11)
12/(1/(13-11))
12*(13-(1*11))
12*((13*1)-11)
12*((13/1)-11)
(12*(13-11))*1
12*((13-11)*1)
12*(13-(11*1))
(12*(13-11))/1
12*((13-11)/1)
12*(13-(11/1))
(13-(1*11))*12
((13*1)-11)*12
((13/1)-11)*12
((13-11)*1)*12
(13-11)*(1*12)
(13-(11*1))*12
((13-11)/1)*12
(13-(11/1))*12
(13-11)/(1/12)
((13-11)*12)*1
(13-11)*(12*1)
((13-11)*12)/1
(13-11)*(12/1)
=====
Waktu yang dibutuhkan: 0.0243989 detik
Simpan ke file [Y/N]: █
```

a. Validasi Karakter Input

Simpan ke file [Y/N]:

b. Validasi Jumlah Input

10 9 8 K

```

Masukkan Input atau Random [1/2]: 1
File atau Keyboard [1/2]: 2
Masukkan kartu yang ingin di cek: A 2 2 2 2
Input is not 4 cards
Masukkan kartu yang ingin di cek: A 10 10 10 10
Input is not 4 cards
Masukkan kartu yang ingin di cek: 10 9 8 K
Input valid
10 9 8 13
=====
98 solusi ditemukan
=====
Solusi:
((10+9)-8)+13
(10+(9-8))+13
10+((9-8)+13)
(10+9)-(8-13)
10+(9-(8-13))
((10+9)+13)-8
(10+9)+(13-8)
(10+(9+13))-8
10+((9+13)-8)
10+(9+(13-8))
((10-8)+9)+13
(10-8)+(9+13)
(10-(8-9))+13
10-(8-(9+13))
10-((8-9)-13)
((10-8)+13)+9
(10-8)+(13+9)
(10-(8-13))+9
10-(8-(13+9))
10-((8-13)-9)
((10+13)+9)-8
(10+13)+(9-8)
(10+(13+9))-8
10+((13+9)-8)
10+(13+(9-8))
((10+13)-8)+9
(10+(13-8))+9
10+((13-8)+9)
(10+13)-(8-9)
10+(13-(8-9))
((9+10)-8)+13
(9+(10-8))+13
9+((10-8)+13)

```

c. Validasi Input Kosong

```
2 10 K J

Masukkan Input atau Random [1/2]: 1
File atau Keyboard [1/2]: 2
Masukkan kartu yang ingin di cek:
Input is empty
Masukkan kartu yang ingin di cek:
Input is empty
Masukkan kartu yang ingin di cek:
Input is empty
Masukkan kartu yang ingin di cek:
Input is empty
Masukkan kartu yang ingin di cek: 2 10 11 J
Masukkan Huruf atau Angka Tidak Valid
Masukkan kartu yang ingin di cek: 2 10 K J
Input valid
2 10 13 11
=====
20 solusi ditemukan
=====
Solusi:
(2+10)*(13-11)
2*((10+13)-11)
2*(10+(13-11))
2*((10-11)+13)
2*(10-(11-13))
2*((13+10)-11)
2*(13+(10-11))
2*((13-11)+10)
2*(13-(11-10))
(10+2)*(13-11)
((10+13)-11)*2
(10+(13-11))*2
((10-11)+13)*2
(10-(11-13))*2
((13+10)-11)*2
(13+(10-11))*2
(13-11)*(2+10)
((13-11)+10)*2
(13-(11-10))*2
(13-11)*(10+2)
=====
Waktu yang dibutuhkan: 0.025082 detik
Simpan ke file [Y/N]: N
Main Lagi? [Y/N]: N
fajarherawan@Fajars-MacBook-Pro bin %
```

#### 4. Input Random

```
Q 5 6 J
SLAMAT DATAN
SAKAWA
Masukkan Input atau Random [1/2]: 2
Q 5 6 J
12 5 6 11
=====
108 solusi ditemukan
=====
Solusi:
((12-5)+6)+11
(12-5)+(6+11)
(12-(5-6))+11
12-(5-(6+11))
12-((5-6)-11)
((12-5)+11)+6
(12-5)+(11+6)
(12-(5-11))+6
12-(5-(11+6))
12-((5-11)-6)

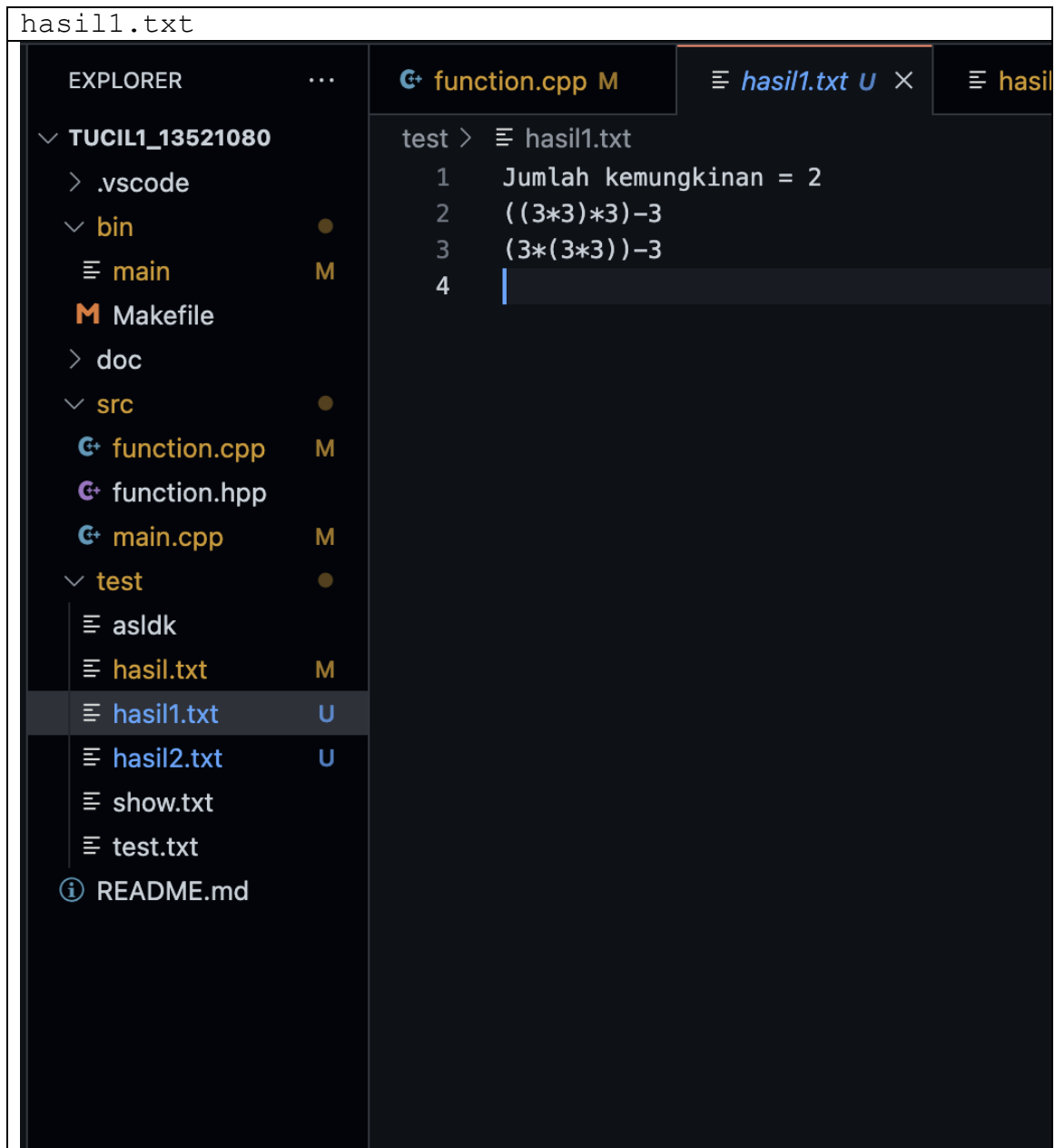
4 12 8 4
=====
Waktu yang dibutuhkan: 0.0241777 detik
Simpan ke file [Y/N]: N
Main Lagi? [Y/N]: Y
Masukkan Input atau Random [1/2]: 2
4 Q 8 4
4 12 8 4
=====
65 solusi ditemukan
=====
Solusi:
(4-12)+(8*4)
4-(12-(8*4))
((4*12)/8)*4
(4*(12/8))*4
4*((12/8)*4)
(4*12)/(8/4)
4*(12/(8/4))
(4-12)+(4*8)
4-(12-(4*8))
```



6 10 J J	
<del>((8*4)+4)-12</del> $(8*4)+(4-12)$ =====	
Waktu yang dibutuhkan: 0.0249126 detik	
Simpan ke file [Y/N]: N	
Main Lagi? [Y/N]: Y	
Masukkan Input atau Random [1/2]: 2	
6 10 J J	
6 10 11 11	
=====	
0 solusi ditemukan	
=====	
Solusi:	
=====	
Q 4 9 A	
Simpan ke file [Y/N]: N	
Main Lagi? [Y/N]: Y	
Masukkan Input atau Random [1/2]: 2	
Q 4 9 A	
12 4 9 1	
=====	
130 solusi ditemukan	
=====	
Solusi:	
$((12+4)+9)-1$	
$(12+4)+(9-1)$	
$(12+(4+9))-1$	
$12+((4+9)-1)$	
$12+(4+(9-1))$	
$(12/4)*(9-1)$	

## 5. Save File

```
Masukkan Input atau Random [1/2]: 1
File atau Keyboard [1/2]: 2
Masukkan kartu yang ingin di cek: 3 3 3 3
Input valid
3 3 3 3
=====
2 solusi ditemukan
=====
Solusi:
((3*3)*3)-3
(3*(3*3))-3
=====
Waktu yang dibutuhkan: 0.016697 detik
Simpan ke file [Y/N]: Y
Masukkan nama file: hasil1.txt
Main Lagi? [Y/N]: Y
Masukkan Input atau Random [1/2]: 1
File atau Keyboard [1/2]: 2
Masukkan kartu yang ingin di cek: 4 4 4 4
Input valid
4 4 4 4
=====
6 solusi ditemukan
=====
Solusi:
(4+4)+(4*4)
4+(4+(4*4))
(4+(4*4))+4
4+((4*4)+4)
((4*4)+4)+4
(4*4)+(4+4)
=====
Waktu yang dibutuhkan: 0.0226928 detik
Simpan ke file [Y/N]: Y
Masukkan nama file: hasil2.txt
Main Lagi? [Y/N]: N
fajarherawan@Fajars-MacBook-Pro bin %
```



```
hasil2.txt

EXPLORER
TUCIL1...
> .vscode
  bin
    main M
  Makefile
  doc
  src
    function.cpp M
    function.hpp
    main.cpp M
  test
    asldk
    hasil.txt M
    hasil1.txt U
    hasil2.txt U
    show.txt
    test.txt
  README.md

function.cpp M
hasil2.txt U
hasil1.txt U
show.txt
test.txt
README.md

test > hasil2.txt
1  Jumlah kemungkinan = 6
2  (4+4)+(4*4)
3  4+(4+(4*4))
4  (4+(4*4))+4
5  4+((4*4)+4)
6  ((4*4)+4)+4
7  (4*4)+(4+4)
8
```

## D. Link Repository

[https://github.com/fajarmhrwn/Tucil1\\_13521080](https://github.com/fajarmhrwn/Tucil1_13521080)

### E. Tabel Checklist Spesifikasi Program

Poin	Ya	Tidak
1. Program berhasil dikompilasi tanpa kesalahan	✓	
2. Program berhasil <i>running</i>	✓	
3. Program dapat membaca input / generate sendiri dan memberikan luaran	✓	
4. Solusi yang diberikan program memenuhi (berhasil mencapai 24)	✓	
5. Program dapat menyimpan solusi dalam file teks	✓	